

Available online at www.sciencedirect.com**ScienceDirect**

Procedia Computer Science 100 (2016) 414 – 420

Procedia
Computer Science

Conference on ENTERprise Information Systems / International Conference on Project
MANagement / Conference on Health and Social Care Information Systems and Technologies,
CENTERIS / ProjMAN / HCist 2016, October 5-7, 2016

A telemedicine application using WebRTC

Mário Antunes^{1,2*}, Catarina Silva^{1,3}, Joaquim Barranca¹

¹*School of Technology and Management, Polytechnic Institute of Leiria, Portugal*

²*Center for Research in Advanced Computing Systems, INESC-TEC, University of Porto, Portugal*

³*Center for Informatics and Systems of the University of Coimbra, Portugal*
{mario.antunes,cataina}@ipleiria.pt, 2130068@my.ipleiria.pt,

Abstract

ICT in healthcare businesses has been growing in Portugal in the past few decades. The implementation of large scale information systems in hospitals, the deployment of electronic prescription and electronic patient records applications are just a few examples. Telemedicine is another emergent and widely used ICT solution to smooth the communication between patients and healthcare professionals, by allowing video and voice transfer over the Internet. Although there are several implementations of telemedicine solutions, they usually have some drawbacks, namely: i) too specific for a purpose; ii) based on proprietary applications; iii) require additional software installation; iv) and usually have associated costs.

In this paper we propose a telemedicine solution based on WebRTC Application Programming Interface (API) to transmit video and voice in real time over the Internet, through a web browser. Besides microphone and webcam control, we have also included two additional functionalities that may be useful to both patients and healthcare professionals during the communication, namely i) bidirectional sending files capability and ii) shared whiteboard which allows free drawing. The proposed solution uses exclusively open source software components and requires solely a WebRTC compatible web browser, like Google Chrome or Firefox. We have made two types of tests in healthcare environment: i) a bidirectional patient-doctor communication; ii) and connecting at one end an external USB medical device with an integrated webcam. The results were promising, since they revealed the potential of using WebRTC API to control microphone and webcam in a telemedicine application, as well as the appropriateness and acceptance of the features included.

© 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the organizing committee of CENTERIS 2016

* Corresponding author. Tel.: +000 000 000 000; fax: +000 000 000 000
E-mail address: mario.antunes@ipleiria.pt

Keywords: eHealth; Telemedicine; WebRTC; Healthcare.

1. Introduction

Telemedicine is a hybrid term that means literally to execute remotely medical tasks, by taking advantage mainly of audio and video facilities. Broadly speaking, governments and healthcare institutions (e.g. hospitals and medical centers) are aware on the use of telemedicine to provide a widely access to medical care from all the individuals, especially those that are not near from the main cities and hospitals. European Commission (EC) advocates telemedicine developments as a crucial part of its document “*eHealth Action Plan 2012-2020: Innovative healthcare for the 21st century*” [1]. Regarding this action line, the EC aims to endow European citizens with secure online access to their patient records by 2015 and widely implement telemedicine technologies in Europe by 2020.

A major contribution of ICT is to narrow distances between people, to smooth the communication process and to enhance citizens’ daily routines. ICT technologies in healthcare industry are booming. Telemedicine follows that trend, not only in Portugal but worldwide [2]. The wide set of solutions available for telemedicine by hospitals and healthcare centers, are usually proprietary, need to be installed on a computer, was designed to be used by a particular kind of professionals, may have costs associated and are confined to the use of the video camera and microphone. As an example, a tele-consultancy given by a doctor to a patient using the well-known Skype software implies that this has to be previously installed in the computers. Another example is the remote consultancy in cardio-pediatrics given by highly specialized doctors, in which they analyze in real-time the images provided by a camera installed in an ultrasound probe device [3,4].

In this paper we present a telemedicine solution aiming to overcome some of the shortcomings and weaknesses previously described. The solution is based on WebRTC technology and uses a WebRTC-compatible web browser to transfer video and voice between two peers, in real time (<https://webrtc.org/>). WebRTC is an open source API that allows the setup of a connection between two peers and the control of webcam and microphone devices. That is, there is no need to install additional software besides a web browser. In addition to video and traditional voice communication in real time, we have included two additional features, namely bidirectional files sending and a digital “whiteboard” for free painting. These additional features could be able to improve the telemedicine experience of doctors and patients, as well as the medical procedure.

The rest of the paper is organized as follows. Section 2 presents the fundamentals regarding telemedicine and WebRTC API. Section 3 details the proposed solution and the experimental setup designed to run the tests. Section 4 presents the results obtained with the use of the telemedicine application proposed in this paper. Section 5 concludes the paper and presents future lines of research.

2. Background

2.1. Telemedicine and eHealth

Telemedicine is an old and generic concept. For the World Health Organization (WHO), telemedicine uses ICTs to overcome geographical barriers and increase access to health care services. This is particularly beneficial for rural and underserved communities in developing countries – groups that traditionally suffer from lack of access to health care. Teleconsulting is the use of telemedicine for medical consulting [5]. E-health is the transfer of health resources and health care by electronic means. There are several challenges to the implementation of telemedicine depending on the player in question; the financial level with the Return On Investment (ROI) or the acquisition cost and system maintenance by some providers. At technological level the main challenges are the inadequacy of systems to the particularities of the institutions; the level of service quality to user satisfaction with the possible lack of ability for handling the technologies involved [6].

2.2. WebRTC

WebRTC stands for Web Real-Time Communications and lies on the TCP/IP application layer. It is an emergent technology embedded into compatible browsers that leverages a set of plugin-free APIs that can be used in desktop and mobile browsers, to enable real time peer-to-peer multimedia communication over the web. WebRTC's intent is to communicate peer-to-peer and for that it needs to know and negotiate several details from the peers: peer's external network IP address; network bandwidth; if there is a video and audio feed and its characteristic; if it is behind a NAT service [7]. Some of these communication details are negotiate by a service call "signalling", that is used to exchange session control messages implemented by Session Description Protocol (SDP), to apply network configurations between peers. Signalling service can be implemented by several ways, like WebSocket, Socket.io and XMPP and SIP protocols. WebRTC implements WebSocket protocol and provides full-duplex real time communication channels over a single TCP connection. SDP provides a standard representation of some proprieties of multimedia session such as media capabilities, transport addresses and other related metadata [8].

In a network topology with NAT the public IP addresses discovery is a critical task. Interactive Connectivity Establishment (ICE) is used to establish communication between two peers, designated ICE candidates, which provide information about the IP address and port number from where the data is going to be exchanged through NAT. ICE uses two distinct services to establish a link between WebRTC peers: i) STUN (Session Traversal Utilities for NAT) is a service that respond to a client with his public IP address and port [9]; 2) TURN (Traversal Using Relays around NAT) is a service that relays communication, so in this case, the communication isn't peer-to-peer [10]. A network topology with STUN and TURN services is depicted in Figures 1a) and 1b) respectively. Classic web architecture is based on HTTP request-reply protocol over a TCP/IP client-server communication. WebRTC extends the classical web architecture, by introducing a peer-to-peer communication paradigm between the client's browsers. Regarding security, WebRTC uses Secure Real-Time Protocol (SRTP) and Datagram Transport Layer Security (DTLS) to establish a secure end-to-end communication.

3. Proposed solution

3.1. Architecture

The telemedicine application is web-based. The client connects to an HTTP server through a web browser and access to the main HTML page that uses also JavaScript and CSS. The web page being accessed by the clients implements JavaScript code to enable the connection to the signaling server, which acts as a broker to coordinate the per-to-peer communication between the browsers. After both clients have been connected to the telemedicine application and have been signaled by the signaling server, the communication between both browsers becomes peer-to-peer, as depicted in Fig. 2a). Fig. 2b) illustrates the general architecture but using a STUN service to establish peer-to-peer connection between both clients, in a scenario where these are behind a NAT router.

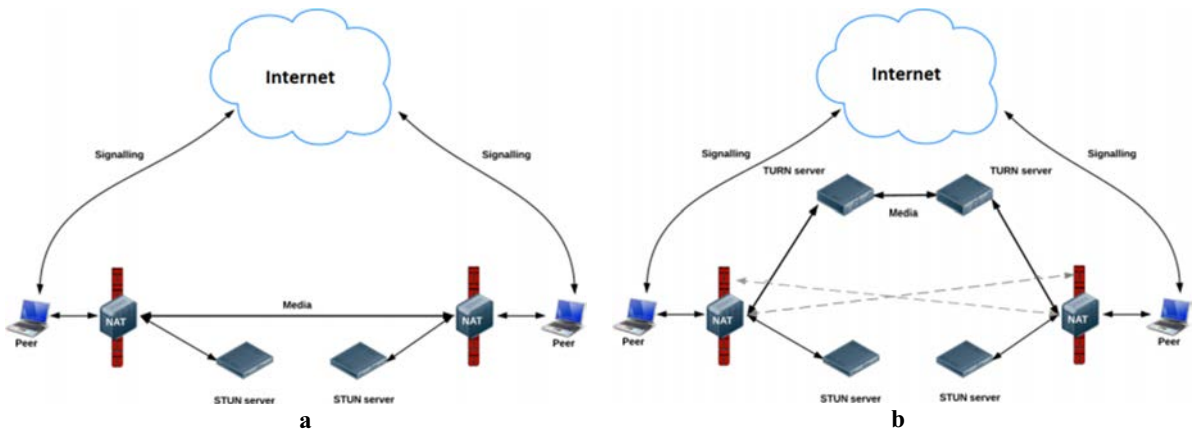


Fig. 1. (a) Network topology with NAT and STUN; (b) Network topology with NAT and TURN.

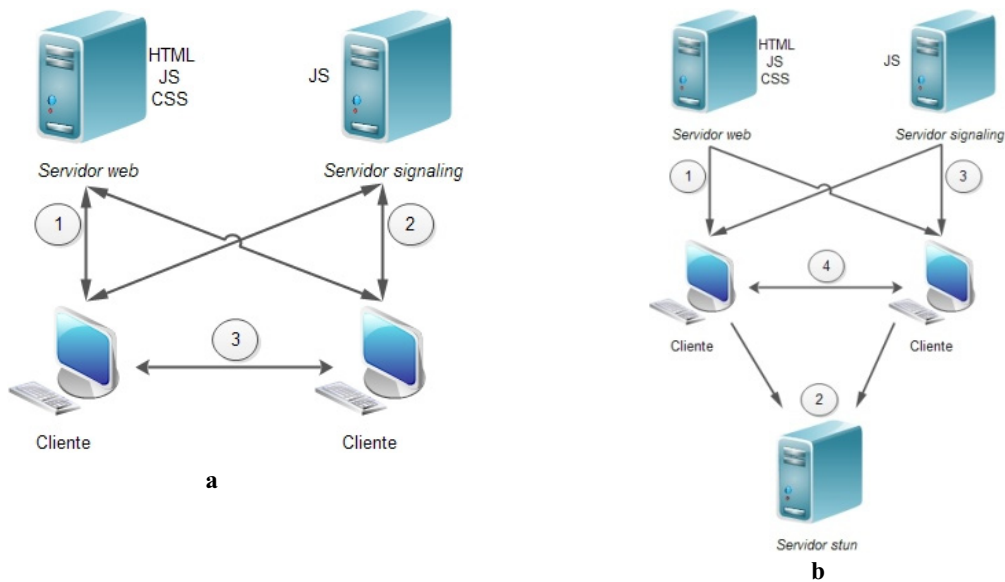


Fig. 2. (a) General architecture; (b) general architecture with STUN.

3.2. Client and server requirements

Each computer (client) that wants to establish a WebRTC connection has to have the following minimum requirements: a web browser, a web cam (internal or external connected through a USB port) and a microphone. At the moment the browsers that support WebRTC are Google Chrome and Mozilla Firefox, being available WebRTC extensions also for Microsoft Internet Explorer and Safari. From the server-side we need to have two servers: i) a web server to host the WebRTC website; ii) and a signaling server to establish and maintain the clients' connection.

The architecture also has to include one or more servers to implement ICE, STUN and TURN services. We have used public servers both for TURN (<http://numb.viagenie.ca>) and STUN (stun1.l.google.com). More recently, we also have installed and configured a standalone TURN server. We installed and tested "reTurn" open-source server (http://resiprocate.org/ReTurn_Overview), which implements both STUN and TURN.

3.3. Development and Web Interface

WebRTC API is designed around three main concepts, *PeerConnection*, *MediaStream* and *DataChannel* [7]. The *PeerConnection* interface represents a WebRTC connection between the local computer and a remote peer. It provides methods to connect to a remote peer, maintain and monitor the connection, and close it once it's no longer needed. The *MediaStream* is responsible for describing a stream of audio or video data, the methods for working with them, the constraints associated with the type of data, the success and error callbacks when using the data asynchronously, and the events that are fired during the process. Finally, the *DataChannel* interface represents a bi-directional data channel between two peers of a connection.

The developed application is divided into some distinct but complementary modules. The core module is responsible for connecting the peers with real time video and audio. It has a session server to identify users that are logged in and available, a real-time collaborative whiteboard area to draw or to annotate over an image and a chat area where users may exchange messages and send files.

Web access requires user identification, composed by a string no less than ten characters that will be made visible to all the users connected to the server. Having gained access and authorized the webcam and microphone control, user will be presented with the interface depicted in Fig. 3. On the left side column we have the users available to initiate a WebRTC connection. In each line corresponding to a user there is a button to start the connection. In that column we may also observe the image provided by the local webcam. After initiating a WebRTC connection with a user, the remote video is shown in the right side column, also there is a button to navigate to the whiteboard area.

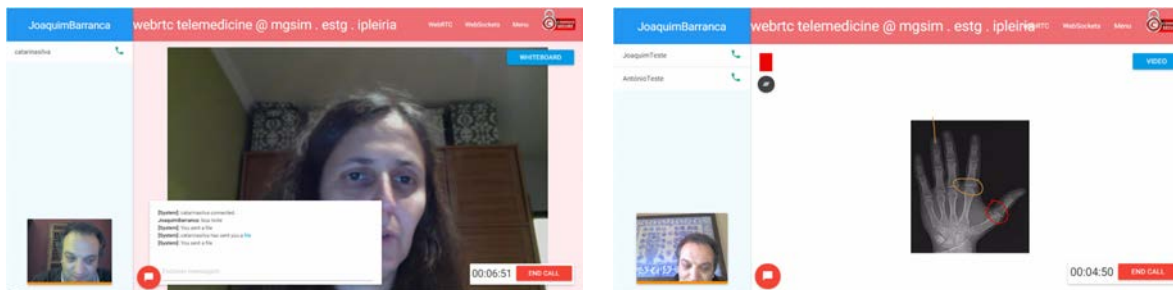


Fig. 3. Interface during a call: (a) Chat and files sending capabilities; (b) Whiteboard with file sharing, free drawing and notes functionalities.

Whiteboard area has three action buttons, namely one to choose the writing color in the upper left of this column, another just below to clear the writing and, at the lower left a chat button. Chat button in the lower left gives access to a double functionality: to send text messages and to send binary files, as depicted in Fig. 3a. Messages and files exchange is made in real time and give the ability to analyze their contents by each user, simultaneously and in a synchronous way. The whiteboard button gives access to a drawing area where it is possible to share drawings in real time or even to draw or to write notes on a previously sent image, like an X-ray file, as illustrated in Fig. 3b. This feature can be particularly interesting to identify in real time, with digital drawing capabilities, a malfunction or anomaly in the previously shared image.

4. Tests and results analysis

4.1. Experimental Setup

The tests were carried out in an experimental setup with the following characteristics:

- Web and signaling servers are running in separated machines, both with Linux Ubuntu Server 14.04 x64, 512 MB RAM and 20 GB SSD.
- Web server is implemented in a Node.js v 0.12.0 instance, supporting HTTPS access over SSL/TLS.

- Web and signaling machines are hosted in the cloud provider Digital Ocean.
- Web browser is Google Chrome v. 49.0.2623.112 (64 bits).
- Telemedicine application is accessed over the following URL: <https://joaquimbarranca.eu/>, registered with the public IP address 46.101.88.57.

4.2. Test 1 – residential clients and Test 2 – healthcare institutions in the RIS network

Test 1 was carried on by two residential users with access to the Internet through a domestic connection provided by the ISPs. We were fully succeeded in this test, being both users able to execute all the functions available in the telemedicine application. Figures 3a) and 3b) depicts the operations made during the WebRTC connection, namely video (and voice) transfer, files send, chat and free drawing in the whiteboard area.

In test 2 users were respectively a doctor specialist in familiar medicine, located at a regional healthcare unit. Both users had distinct hardware configurations and Internet connections. The doctor had a P4 PC with 1GB RAM in a 10 Mbps network, while the patient used a core i3 PC with 4GB RAM in a 100 Mbps network. They were also connecting to the Internet through the network that connects all the hospitals and healthcare institutions in Portugal, named “Rede Informática da Saúde” (RIS). We experienced some network configuration problems as their public IP addresses were not able to be identified by ICE servers. This issue is probably related with the RIS network configuration, specifically with its NAT and firewall configurations. A subsequent test with a similar configuration had similar results. In this case the doctor was located at a Central Hospital and the patient was at home.

4.3. Test 3 – external USB medical device with incorporated camera

In the last test we analysed the usability and feasibility of connecting an external device with an integrated camera. We used a portable USB digital microscope that, after connected to the USB port, became able to send video signal to the remote peer through the Internet (Fig. 4). The monitor is split into two parts, each with the image sent by each peer connection. Regarding the microscopic device, the figure also illustrates the professional pointing the microscope to the harm, which is the image sent to the other peer connection though WebRTC.

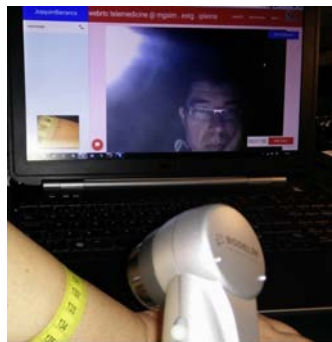


Fig. 4. Test with video, web camera, chat and files sending.

5. Conclusions and future work

The purpose of this paper was to present a telemedicine solution that could smooth the remote access to healthcare by all the patients, particularly those that live in remotely areas and far from the central hospitals. We proposed a WebRTC-based solution that uses web communication to send voice and video between the peers. Besides microphone and camera, we have also included in the application a bidirectional sending files option and a whiteboard interface. The tests carried on revealed the benefits of the implemented features, as well as the exchange of video between an external device with a camera and a healthcare professional. We have experienced problems in

some tests realized between users that were communicating from or to the hospital computers network (RIS). This is probably due to networking configuration issues that should be overcome with more intensive tests. According to our knowledge, there is not actually a telemedicine solution using WebRTC technology that was provided by Portuguese healthcare institutions. Regarding its low cost and easy to install and setup, we aim to strengthen the cooperation and involvement with Portuguese healthcare institutions and developers, in order to intensify the tests toward an effective implementation in hospitals and medical centers.

References

- [1] Commission, E. (2012). “Putting patients in the driving seat a digital future for healthcare”.
- [2] Peña-López, I. et al. (2010). “Improving health sector efficiency: The role of information and communication technologies”.
- [3] Mendelson, M.; Vivodtzev, I.; Tamisier, R.; Laplaud, D.; Domingos, S. D.; Baguet, J.; Moreau, L.; Koltes, C.; Pepin, L.; Chavez, G. (2014). Continuous positive airway pressure (cpap) supported by telemedicine improves sleepiness and quality of life but not blood pressure in high cardiovascular risk obstructive sleep apnea (osa): A randomized, controlled trial. *European Respiratory Journal*, 42.Suppl 57, pp.P2044.
- [4] Qiang, J. K. and Marras, C. (2015). “Telemedicine in parkinson’s disease: A patient perspective at a tertiary care centre”. *Parkinsonism & Related Disorders*.
- [5] Organization-WHO, W. H. (2011). “Atlas ehealth country profiles based on the findings of the second global survey on eHealth”.
- [6] Hjelm, N. (2005). Benefits and drawbacks of telemedicine. *Journal of tele-medicine and telecare*, 11(2):60–70.
- [7] Johnston, A., and Burnett, C.; “WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web”. Digital Codex LLC; 2012
- [8] Handley, M.; Colin P.; and Jacobson, V. “SDP: session description protocol.”; RFC 4566 (Proposed Standard); 2006
- [9] Rosenberg, J.; Mahy, R.; Matthews, P.; Wing, D. “Session traversal utilities for NAT (STUN)”.; RFC 5389 (Proposed Standard); 2008
- [10] Matthews, P.; Mahy, R.; Rosenberg, J. “Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun)”.; RFC 5766 (Proposed Standard); 2010